

Writing on Dirty Flash Memory

Yongjune Kim and B. V. K. Vijaya Kumar

Department of Electrical & Computer Engineering, Data Storage Systems Center (DSSC)
Carnegie Mellon University
Pittsburgh, PA, USA

Email: yongjunekim@cmu.edu, kumar@ece.cmu.edu

Abstract—The most important challenge in the scaling down of flash memory is its increased inter-cell interference (ICI). If side information about ICI is known to the encoder, the flash memory channel can be viewed as similar to Costa’s “writing on dirty paper (dirty paper coding).” We first explain why flash memories are *dirty* due to ICI. We then show that “dirty flash memory” can be changed into “memory with defective cells” model by using only one pre-read operation. The asymmetry between write and erase operations in flash memory plays an important role in this change. Based on the “memory with defective cells” model, we show that additive encoding can significantly improve the probability of decoding failure by using the side information.

I. INTRODUCTION

Aggressive scaling down of memory cell size has driven the continuous growth of flash memory density. However, the scaling down leads to many challenges. One primary challenge is the increased inter-cell interference (ICI) between adjacent (neighboring) flash memory cells [1]. As the distance between adjacent cells decreases due to scaling down, flash memory cells suffer from higher ICI [1], [2].

In order to cope with the ICI, various approaches have been proposed. Device level approaches such as new materials and novel cell structures try to reduce the parasitic capacitances between adjacent cells [3]. At circuit and architecture levels, several write (program) schemes and all bitline (ABL) architecture were proposed to deal with the ICI [4]–[6].

Recently, strong error control codes (ECC) such as low-density parity check (LDPC) codes and signal processing have been also investigated [7]–[9]. The disadvantage of ECC with soft decision decoding and signal processing is the degradation of read speed due to multiple reads needed to obtain the soft decision values. In addition, modulation coding has been investigated to reduce some data patterns which are vulnerable to ICI [10], [11]. The significant redundancy of modulation coding is an important drawback.

In this paper, we propose a scheme that uses the side information corresponding to the ICI. In particular, the encoder uses this side information to improve the decoding failure probability, but at the expense of decreased write speed. The decrease in write speed may be acceptable for memory systems, since the write operation is typically not on the critical path because of write buffers available in the memory hierarchy [12], [13]. In addition, the read speed degradation would be more critical in applications such as one-time programmable (OTP) flash memories.

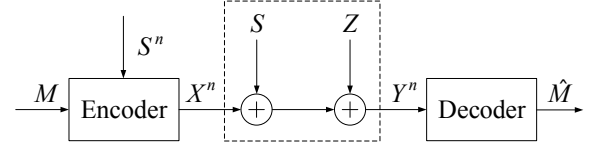


Fig. 1. Dirty paper channel where M and \hat{M} represent a message and its estimate, respectively [17].

Theoretically, the proposed scheme can be explained by Gelfand-Pinsker problem. The Gelfand-Pinsker problem assumes that only the encoder knows noncausally the side information of the channel [14]. There are two famous examples of Gelfand-Pinsker problem: “Writing on dirty paper (dirty paper coding)” in *communication* and “memory with defective cells” in *storage* [15]–[17].

Costa’s writing on dirty paper considers the following channel [15]:

$$Y = X + S + Z \quad (1)$$

where X and Y are the channel input and output, respectively. Also, $S \sim N(0, \sigma_S^2)$ is an interference and $Z \sim N(0, \sigma_Z^2)$ is an additive noise. Assume that the channel input satisfies an average power constraint $\frac{1}{n} \sum_{i=1}^n X_i^2 \leq P$ for the channel input vector $X^n = (X_1, \dots, X_n)$. If neither the encoder nor the decoder knows S , the capacity is given by

$$C_{\min} = \frac{1}{2} \log_2 \left(1 + \frac{P}{\sigma_S^2 + \sigma_Z^2} \right). \quad (2)$$

As shown in Fig. 1, if the encoder knows the entire interference vector $S^n = (S_1, \dots, S_n)$ prior to transmission, the capacity is given by

$$C_{\max} = \frac{1}{2} \log_2 \left(1 + \frac{P}{\sigma_Z^2} \right) \quad (3)$$

where the effect of the interference S is completely cancelled out [15].

The memory with defective cells was introduced by Kuznetsov and Tsybakov [16]. A binary memory cell is called defective if its cell value is stuck-at a particular value regardless of the channel input. As shown in Fig. 2, this channel model has a ternary defect information $S^+ \in \{0, 1, \lambda\}$ whereas the channel input X and output Y are binary. The state $S^+ = 0$ corresponds to a stuck-at 0 defect that always outputs a 0 independent of its input value, the state $S^+ = 1$ corresponds to a stuck-at 1 defect that always outputs a 1, and the state $S^+ = \lambda$ corresponds to a normal cell that

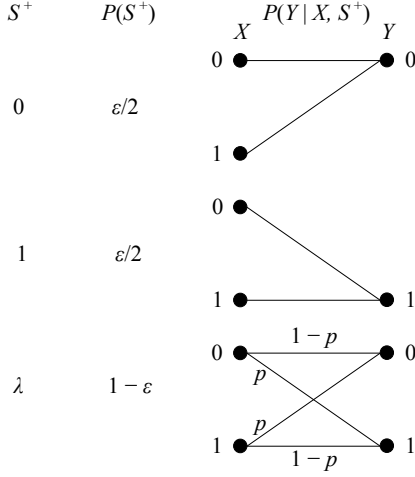


Fig. 2. Memory with defective cells.

can be modelled by a binary symmetric channel (BSC) with crossover probability p . The probabilities of these states are $\varepsilon/2, \varepsilon/2$ (assuming a symmetric defect probability), and $1 - \varepsilon$, respectively [18].

If neither the encoder nor the decoder knows S^+ , the capacity is given by

$$C_{\min}^+ = 1 - h\left((1 - \varepsilon)p + \frac{\varepsilon}{2}\right) \quad (4)$$

where $h(x) = -x \log_2 x - (1 - x) \log_2 (1 - x)$. Note that (4) equals the capacity of a BSC with crossover probability $\tilde{p} = (1 - \varepsilon)p + \frac{\varepsilon}{2}$. If the encoder or the decoder knows $(S^+)^n = (S_1^+, \dots, S_n^+)$ of n memory cells, the maximum capacity of memory with defective cells can be achieved [18]. The capacity is given by

$$C_{\max}^+ = (1 - \varepsilon)(1 - h(p)). \quad (5)$$

The common ground of writing on dirty paper and memory with defective cells is that in both these cases only the encoder knows S^n or $(S^+)^n$. Thus, these two examples can be categorized as Gelfand-Pinsker problem.

Suppose that S^n represents the ICI of n flash memory cells. Then, the important question is how the encoder knows S^n accurately. Unfortunately, it is difficult for the encoder to know S^n due to flash memory's properties (the details are explained in Section III-A). Thus, we change flash memory channel with the ICI into flash memory with defective cells, which means that the encoder uses the side information of defects $(S^+)^n$ rather than the side information of ICI S^n . The asymmetry between write and erase operations of flash memory plays a pivotal role in this change (the asymmetry between write and erase operations is explained in Section II-A). It is worth mentioning that two famous examples of Gelfand-Pinsker problem come together in flash memories.

After changing the flash memory channel with the ICI into the model of memory with defective cells, we consider the additive encoding to use the side information $(S^+)^n$ [18], [19]. The numerical results show that the proposed scheme

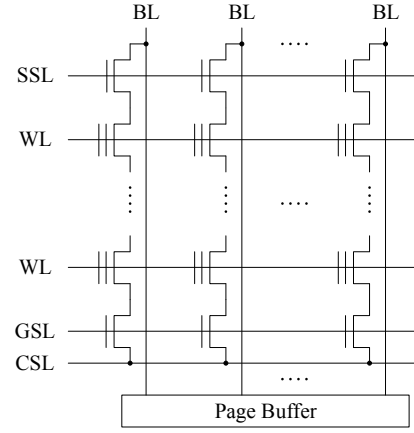


Fig. 3. Flash memory block where SSL, GSL, and CSL denote string select line, ground select line, and common source line, respectively [21].

can improve the probability of decoding failure significantly due to $(S^+)^n$ available at the encoder.

Recently, coding schemes that use the side information at the encoder have drawn attention for phase change memories (PCM) and write once memory (WOM) codes [12], [20]. We will focus on the ICI of flash memories and propose a scheme to obtain the side information corresponding to the ICI efficiently and improve the probability of decoding failure by using this information at the encoder.

The rest of this paper is organized as follows. Section II reviews the basics of flash memory such as flash memory structure, asymmetry between write and erase operations, and ICI. Section III explains why flash memories are dirty and how it can be formulated into the model of memory with defective cells. Also, the additive encoding for this model will be briefly explained. After showing the numerical results in Section IV, Section V concludes the paper.

II. BACKGROUND

A. Flash Memory Basics

Each flash memory cell is a floating gate transistor whose threshold voltage can be configured by controlling the amount of electron charges in the floating gate [9]. More electrons in the floating gate make the corresponding cell's threshold voltage higher.

As shown in Fig. 3, each flash memory block is a two-dimensional cell array where each cell is connected to a wordline (WL) and a bitline (BL). For B -bit per cell flash memory, each WL stores B -page data.

In order to store B -bit per cell, each cell's threshold voltage is divided into 2^B states, which is similar to pulse amplitude modulation (PAM). Fig. 4 (a) shows the threshold voltage distribution of 1-bit per cell flash memory, which is traditionally called single-level cell (SLC). Initially, all memory cells are erased, so their threshold voltage is in the lowest state S_0 . In order to store data, some of cells in S_0 should be written (programmed) into S_1 . For multi-level cell (MLC) flash memories (i.e., $B \geq 2$), some of cells in S_0 (erase state)

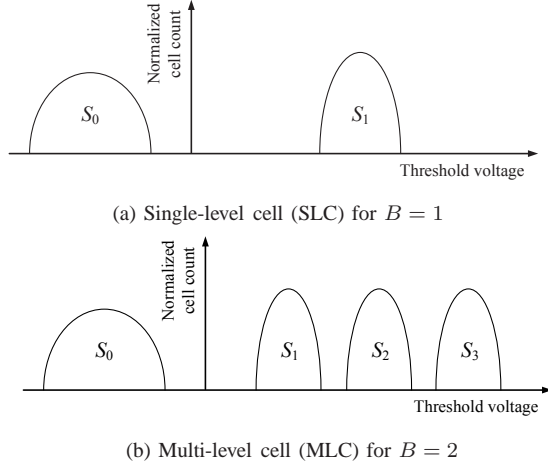


Fig. 4. Threshold voltage distribution of flash memory cells.

will be written into S_1, \dots, S_{2^B-1} (program states) as shown in Fig. 4 (b).

The most widely used write operation scheme is the incremental step pulse programming (ISPP) scheme, which was proposed to maintain a tight threshold voltage distribution for high reliability [21]. The ISPP is based on repeated program and verify cycles with the staircase program voltage V_{pp} . Each program state associates with a verify level that is used in the verify operation. During each program and verify cycle, the floating gate threshold voltage is boosted by up to the incremental step voltage ΔV_{pp} and then compared with the corresponding verify level. If the threshold voltage of the memory cell is still lower than the verify level, the program and verify iteration continues. Otherwise, further programming of this cell is disabled [9], [21].

The positions of program states are determined by verify levels and the tightness of each program state depends on the incremental step voltage ΔV_{pp} . By reducing ΔV_{pp} , the threshold voltage distribution can be made tighter, however the write time increases [21], [22].

In read operation, the threshold voltages of cells in the same WL are compared to a given read level. After a read operation, a page of binary data is transferred to the page buffer in Fig. 3. The binary data shows whether the threshold voltage of each cell is lower or higher than the given read level. Namely, the read operation of flash memory is a binary decision. Thus, multiple read operations are required to obtain a soft decision value, which lowers the read speed. The degradation of read speed is an important challenge for soft decision decoding and signal processing [7], [9].

The threshold voltage of flash memory cell can be reduced by erase operation. In flash memory, all the memory cells in the same flash memory block should be erased at the same time [21]. Note that a page of data (within a WL) can be written or read (generally, a flash memory block consists of 64 WLs [6]). In addition, the threshold voltage of cell should be moved into the lowest state S_0 by erase operation whereas a slight increase of threshold voltage is possible by ISPP during write operation [21]. These unique properties of flash memory

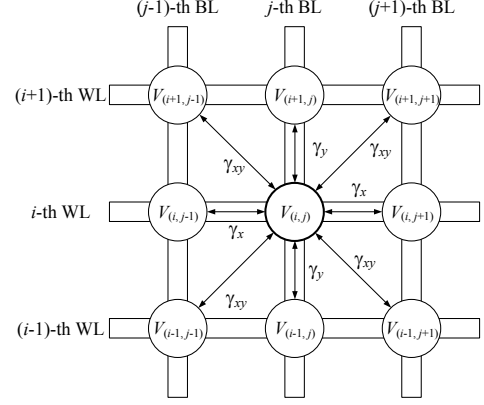


Fig. 5. Inter-cell interference between adjacent cells.

cause *asymmetry between write and erase operations*.

B. Inter-cell Interference

In flash memory, the threshold voltage shift of one cell affects the threshold voltage of its adjacent cell because of the ICI. The ICI is mainly attributed to parasitic capacitances coupling effect between adjacent cells [1], [2].

Fig. 5 illustrates the ICI between adjacent cells. $V_{(i,j)}$ is the threshold voltage of (i,j) cell which is situated at i -th WL and j -th BL. γ_x is x -directional coupling ratio between BL and adjacent BL. Also, γ_y is y -directional coupling ratio between WL and adjacent WL. Finally, γ_{xy} is xy -directional (diagonal) coupling ratio. These coupling ratios depend on parasitic capacitances between adjacent cells. As the cell size continues to shrink, the distances between cells become smaller, which results in the increase of the parasitic capacitances. The increase of parasitic capacitances causes the increase of coupling ratios [1], [2].

According to [2], the threshold voltage shift $\Delta_{ICI}V_{(i,j)}$ of (i,j) cell due to the ICI is given by

$$\begin{aligned} \Delta_{ICI}V_{(i,j)} = & \gamma_x (\Delta V_{(i,j-1)} + \Delta V_{(i,j+1)}) \\ & + \gamma_y (\Delta V_{(i-1,j)} + \Delta V_{(i+1,j)}) \\ & + \gamma_{xy} (\Delta V_{(i-1,j-1)} + \Delta V_{(i-1,j+1)} + \\ & \Delta V_{(i+1,j-1)} + \Delta V_{(i+1,j+1)}) \end{aligned} \quad (6)$$

where $\Delta V_{(i\pm 1,j\pm 1)}$ in the right hand side represent the threshold voltage shifts of adjacent cells after the (i,j) cell has been written. The ICI that happens before writing (i,j) cell can be compensated by several write schemes so long as (i,j) cell is in program states [4], [5]. Note that the ICI to (i,j) cell in S_0 cannot be compensated by these write schemes since a cell in S_0 is never written (i.e., stay in S_0) [5], [11].

III. WRITING ON DIRTY FLASH MEMORY

A. Flash Memory Channel

The flash memory channel can be given by

$$Y = X + S + Z \quad (7)$$

$$= X + Z_{\text{write}} + S + Z_{\text{read}} \quad (8)$$

$$= V + S + Z_{\text{read}} \quad (9)$$

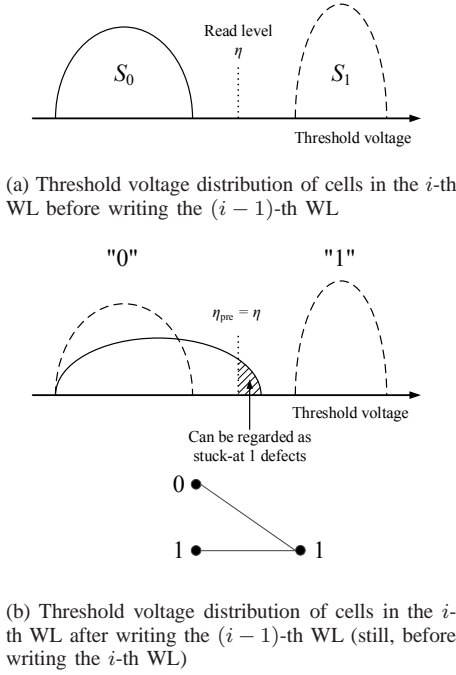


Fig. 6. Change from the flash memory channel with the ICI to the model of memory with defective cells by one pre-read operation.

where X and Y are the channel input and output. Also, S represents the ICI from adjacent cells. The additive random noise Z is a sum of Z_{write} and Z_{read} where Z_{write} is the write noise due to the initial threshold voltage distribution after erase operation and the incremental step voltage ΔV_{pp} of ISPP. Z_{read} is the read noise due to other noise sources.

Since the write noise Z_{write} precedes the ICI S , we consider a random variable $V = X + Z_{\text{write}}$. As shown in (6), the shifts of V in adjacent cells determine the ICI S . Thus, we claim that the ICI S of (i, j) cell is the same as $\Delta_{\text{ICI}} V_{(i, j)}$ of (6). The read noise Z_{read} happens after ICI. The channel model of (7) was validated by the real data from the 2x nm NAND flash memory [23].

It seems that (7) is the same as (1). However, there are important differences between the flash memory channel and the dirty paper channel of (1). First, S of (7) depends on the adjacent cells' X and Z_{write} whereas S of (1) are independent. In addition, the mean of S in (7) is not zero since the coupling ratios $(\gamma_x, \gamma_y, \gamma_{xy})$ are positive and the threshold voltage shifts of adjacent cells $\Delta V_{(i\pm 1, j\pm 1)}$ in (6) are nonnegative.

Now, we discuss why it is difficult for the encoder to know S . First, the encoder has to know the channel input X of adjacent cells in different WLs. Since the write operation is performed page by page, it is possible for the encoder to know the channel input of cells in several WLs only the case where a large number of continuous pages are written at a time [9].

Even in the case where the encoder knows enough channel input X of several WLs in advance, it is still difficult to know the random variable $V = X + Z_{\text{write}}$ that determines the ICI. Note that the write noise Z_{write} is random. In addition, it is much more complicated to know the voltage shift of adjacent cells (i.e., $\Delta V_{(i\pm 1, j\pm 1)}$ in (6)) since flash memory's

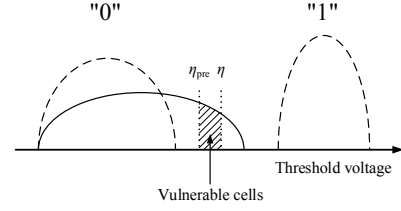


Fig. 7. Vulnerable cells can also be regarded as stuck-at 1 defects by setting a pre-read level such that $\eta_{\text{pre}} < \eta$.

read operation is inherently binary decision. Thus, multiple read operations are required to know $\Delta V_{(i\pm 1, j\pm 1)}$.

B. Dirty Flash Memory

We describe how to change the flash memory channel with the ICI into the model of memory with defective cells. By this change, the encoder can readily obtain the side information of defects S^+ instead of the side information of ICI S .

Consider an SLC flash memory. Fig. 6 shows the threshold voltage distribution of cells in the i -th WL before writing. Initially, all cells are in the erase state S_0 as shown in Fig. 6 (a). However, after writing the adjacent $(i-1)$ -th WL, the threshold voltages of cells in the i -th WL will be distorted due to the ICI from the $(i-1)$ -th WL as shown in Fig. 6 (b). Thus, some of cells' threshold voltages can be higher than the given read level η though the i -th WL has yet to be written.

As explained in Section II-A, the threshold voltage of flash memory cells cannot be reduced during write operation. In order to decrease the threshold voltage of a cell, we have to erase the whole flash memory block. Thus, the cell whose threshold voltage is higher than the read level η will be decided as S_1 . Assume that S_0 and S_1 denote the data "0" and "1" respectively. If a "0" is attempted to be written to this cell, an error results. However, "1" can be written into this cell. Thus, these cells can be regarded as stuck-at 1 defects in Fig. 2. Even if some of the cells which are regarded as stuck-at 1 defects may be "0" due to the read noise Z_{read} , this error can be corrected by the partitioned linear block codes (PLBC) explained in Section III-C.

The defect information of stuck-at 1 defects, i.e., S^+ can be obtained by just one read operation before writing the i -th WL. Before writing the i -th WL, the read operation will be performed at the given read level, i.e., *pre-read operation*. When the read level for the pre-read operation, i.e., *pre-read level* η_{pre} is the same as the read level η , the cells whose threshold voltages are higher than the read level η can be identified by the pre-read operation. Thus, the encoder can know the side information of defects S^+ .

Let \circ denote the operator $\circ : \{0, 1\} \times \{0, 1, \lambda\} \rightarrow \{0, 1\}$ by

$$x \circ s^+ = \begin{cases} x, & \text{if } s^+ = \lambda; \\ s^+, & \text{if } s^+ \neq \lambda \end{cases} \quad (10)$$

where $x \in X$ and $s^+ \in S^+$. The binary channel input X and the ternary defect information S^+ are shown in Fig. 2 [24].

By only one pre-read operation before writing, the flash memory channel with the ICI in (7) can be changed into the following channel model of *binary* memory with defective cells.

$$Y = X \circ S^+ + Z \quad (11)$$

where X , Y , and Z are the *binary* channel input, output, and additive noise, respectively. In contrast, X , Y , and Z of (7) are *real* values. Note the difference between “ $X + S$ ” in (7) and “ $X \circ S^+$ ” in (11).

It is worth mentioning that S^+ does not reveal the x -directional ICI from the i -th WL and the ICI from the $(i+1)$ -th WL, which are subsequent ICI since the pre-read operation is done before the write operations of i -th and $(i+1)$ -th WLs.

However, the effect of these subsequent ICI can be alleviated by changing the pre-read level. Suppose that the pre-read level η_{pre} is lower than the read level η as shown in Fig. 7. A cell whose threshold voltage is between η_{pre} and η is a vulnerable cell though it is not a stuck-at 1 defect. When the data “0” is written to this cell, the ISPP cannot change the threshold voltage of this cell and its threshold voltage is near η . Thus, it is vulnerable to the subsequent ICI and read noise. On the other hand, the cell’s threshold voltage will be higher than a verify level of S_1 by the ISPP when the data “1” is written to this cell. Note that the verify level of S_1 is higher than the read level η .

Thus, by setting a pre-read level such that $\eta_{\text{pre}} < \eta$, we can regard all the cells whose threshold voltages are higher than the pre-read level η_{pre} as stuck-at 1 defects. Using the additive encoding, only the data “1” will be written to these cells. Thus, we can obtain more noise margin between S_0 and S_1 and prevent the subsequent ICI and read noise.

Imagine a sheet of *lined* paper (Costa considered a sheet of *blank* paper in [15]). A flash memory block is a sheet of paper and each WL corresponds to a row between lines. If a row between lines is spacious, then the writer can easily write a message between lines. In order to write more messages on a sheet of paper, the writer tries to narrow the space between lines (i.e., scaling down). However, as the space between lines narrows, it is more difficult to write a message without crossing the lines (i.e., ICI). Eventually, after writing a message in a narrower space, the adjacent rows have more dirty spots (i.e., stuck-at 1 defects) due to the ink marks crossing the line. One way to solve this problem is to erase the dirty spots in a corresponding row before writing. However, erasing a row is not permitted (because of the asymmetry between write and erase in flash memory).

Now we consider the other way instead of erasing a row before writing. Assume that the writer knows the location of the dirty spots, but the reader cannot distinguish between the message and the dirt [15]. Hence, the problem of writing on flash memory with the ICI can be considered as a Costa’s writing on dirty paper, i.e., *writing on dirty flash memory*. Since the dirty spots are changed into stuck-at 1 defects by the pre-read operation, writing on dirty flash memory is equivalent to writing on (flash) memory with defective cells.

Thus, “writing on dirty paper” in communication and “memory with defective cells” in storage come together in flash memory.

C. Writing on Dirty Flash Memory

Now the encoder knows the side information of stuck-at 1 defects of flash memory, i.e., the writer knows the dirty spots on a sheet of lined paper. The next step is to write a message on dirty flash memory taking into account that the decoder cannot distinguish between the 1s in a message and the stuck-at 1 defects, i.e., the reader cannot distinguish between the ink marks and the dirty spots.

Tsybakov proposed the *additive encoding* approach which masks defects by adding a carefully selected binary vector [19], [25]. Masking defects is to make a codeword whose values at the locations of defects match the stuck-at values at those locations. Heegard elaborated the additive encoding and defined the $[n, k, l]$ partitioned linear block codes (PLBC) that mask stuck-at defects and correct random errors [24].

We will apply the $[n, k, l]$ PLBC for writing on dirty flash memory. A vector version of (11) for an n -cell memory is given by

$$\mathbf{y} = \mathbf{x} \circ \mathbf{s}^+ + \mathbf{z} \quad (12)$$

where $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \{0, 1\}^n$ are the binary channel input vector, output vector, and random error vector, respectively. Also, $\mathbf{s}^+ \in (S^+)^n$ represents the side information of defect locations and stuck-at values. Both \circ and $+$ are the vector component-wise operators.

The number of defects in n cells is equal to the number of non- λ components in \mathbf{s}^+ . The number of errors due to defects is given by

$$\|\mathbf{x} \circ \mathbf{s}^+ - \mathbf{x}\| \quad (13)$$

where $\|\cdot\|$ is the Hamming weight of the vector.

The $[n, k, l]$ PLBC is a pair of linear subspaces $\mathcal{C}_1 \subset \{0, 1\}^n$ and $\mathcal{C}_0 \subset \{0, 1\}^n$ of dimension k and l such that $\mathcal{C}_1 \cap \mathcal{C}_0 = \{0\}$. The encoding and decoding of the $[n, k, l]$ PLBC are summarized as follows. (The details were presented in [24].)

Encoding: A message $\mathbf{m} \in \{0, 1\}^k$ is encoded to a corresponding codeword $\mathbf{c} \in \mathcal{C}$ as follows.

$$\mathbf{c} = G_1 \mathbf{m} + G_0 \mathbf{d} = [G_1 \ G_0] \begin{bmatrix} \mathbf{m} \\ \mathbf{d} \end{bmatrix} \quad (14)$$

$$= \tilde{G} \begin{bmatrix} \mathbf{m} \\ \mathbf{d} \end{bmatrix} \quad (15)$$

where $\mathbf{c}_1 = G_1 \mathbf{m} \in \mathcal{C}_1$ and $\mathbf{c}_0 = G_0 \mathbf{d} \in \mathcal{C}_0$. Note that $\mathbf{d} \in \{0, 1\}^l$ is the parity for masking defects. The generator matrix G_1 is an $n \times k$ matrix and the generator matrix G_0 is an $n \times l$ matrix. Thus, \mathcal{C} can be regarded as an $[n, k+l]$ linear block code with the generator matrix $\tilde{G} = [G_1 \ G_0]$.

Decoding: Retrieve $\mathbf{y} = \mathbf{x} \circ \mathbf{s}^+ + \mathbf{z}$ where $\mathbf{x} = \mathbf{c}$. Compute the syndrome $\mathbf{w} = \tilde{H}^T \mathbf{y}$ (superscript T denotes transpose) and choose $\hat{\mathbf{z}} \in \{0, 1\}^n$ which minimizes $\|\mathbf{z}\|$ subject to $\tilde{H}^T \mathbf{z} = \mathbf{w}$. Then $\hat{\mathbf{m}} = \tilde{G}_1^T \hat{\mathbf{c}}$ where $\hat{\mathbf{c}} = \mathbf{y} + \hat{\mathbf{z}}$. The parity check matrix \tilde{H} is an $n \times r$ matrix such that $\tilde{H}^T \tilde{G} = \mathbf{0}_{r, k+l}$ (the $r \times (k+l)$ zero matrix) and $k + l + r = n$. The message inverse matrix

TABLE I
SIMULATION PARAMETERS

Parameters	Values
Bits per cell	$B = 1$ (SLC)
Architecture	All bitline (ABL)
Initial threshold voltage distribution	$\mathcal{N}(-3, 1^2)$
Verify level for S_1	$V_{S_1} = 1$
Incremental step voltage	$\Delta V_{pp} = 1$
Coupling ratios $(\gamma_x, \gamma_y, \gamma_{xy})$	$\alpha (0.08, 0.1, 0.006)$
Z_{read} of (9)	$\mathcal{N}(0, \sigma_{Z_{\text{read}}}^2)$
Read level between S_0 and S_1	$\eta = 0$
Additive encoding	$[n = 1023, k = 923, l]$ PBCH Codes

TABLE II
ALL POSSIBLE REDUNDANCY ALLOCATION CANDIDATES OF
 $[n = 1023, k = 923, l]$ PBCH CODES

Code	l	r	Notes
0	0	100	Only correcting random errors
1	10	90	
2	20	80	
3	30	70	
4	40	60	
5	50	50	
6	60	40	
7	70	30	
8	80	20	
9	90	10	
10	100	0	Only masking defects

\tilde{G}_1 is defined as an $n \times k$ matrix such that $\tilde{G}_1^T G_1 = I_k$ (the k -dimensional identity matrix) and $\tilde{G}_1^T G_0 = 0_{k,l}$.

The parity \mathbf{d} for masking defects determines the binary vector \mathbf{c}_0 masking stuck-at defects. The encoder should choose \mathbf{d} judiciously by considering both \mathbf{c}_1 and \mathbf{s}^+ . The optimal \mathbf{d} is chosen to minimize the number of errors due to defects, i.e. $\|\mathbf{c} \circ \mathbf{s}^+ - \mathbf{c}\|$. Since the computational complexity for finding the optimal \mathbf{d} is exponential, we use the *two-step encoding scheme* for determining \mathbf{d} which was proposed in [26], [27].

If $\|\mathbf{c} \circ \mathbf{s}^+ - \mathbf{c}\| \neq 0$, there are errors due to unmasked defects. Since \mathbf{S}^+ contains partial information of \mathbf{S} , the remaining ICI which is not included in \mathbf{S}^+ results in errors. Also, we should consider the random errors in cells (even the cells regarded as stuck-at 1 defects) due to the read noise Z_{read} in (9). All these errors will be regarded as random errors during decoding.

IV. NUMERICAL RESULTS

In this section, we present the numerical results when the encoder uses the side information of \mathbf{s}^+ in (12). The simulation parameters are summarized in Table. I. The initial threshold voltage distribution (after erasing a flash memory block) is assumed to be the Gaussian distribution $\mathcal{N}(-3, 1^2)$. The ISPP

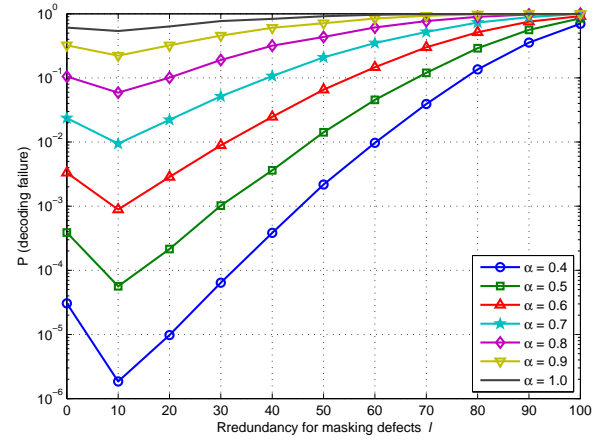


Fig. 8. Comparison of $P(\text{decoding failure})$ for different scaling factors α ($\sigma_{Z_{\text{read}}} = 0.1$, $\eta_{\text{pre}} = \eta = 0$).

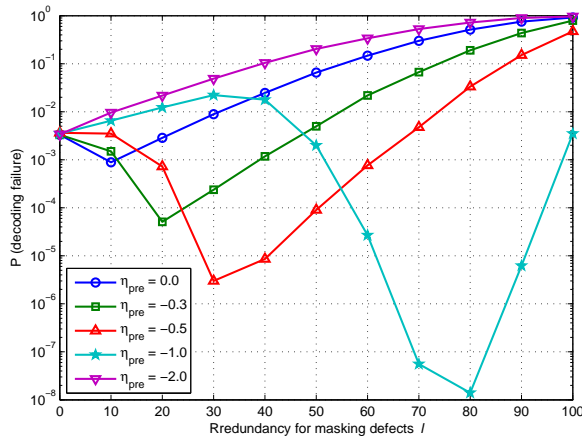
was implemented with the parameters of the verify level for S_1 , i.e., $V_{S_1} = 1$ and the incremental step voltage $\Delta V_{pp} = 1$. Note that the variance of initial threshold voltage distribution and the incremental step voltage work for Z_{write} of (8), which precedes the ICI.

The ICI S is calculated by (6) where the coupling ratios are given by $(\gamma_x, \gamma_y, \gamma_{xy}) = \alpha (0.08, 0.1, 0.006)$. The scaling factor α represents the ICI strength, and the ratios between γ_x , γ_y , and γ_{xy} are taken from [1]. These ratios can be different for each product of flash memory. The read noise Z_{read} after the ICI is assumed to the $\mathcal{N}(0, \sigma_{Z_{\text{read}}}^2)$.

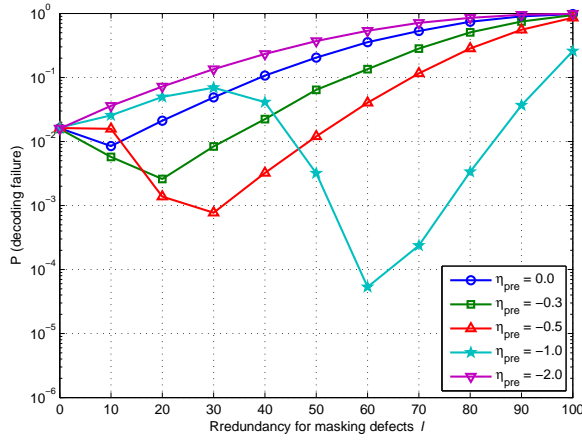
For additive encoding, we consider $[n = 1023, k = 923, l]$ partitioned Bose, Chaudhuri, Hocquenghem (PBCH) codes. The PBCH code is a special class of PLBC, which can be designed by a similar method of standard BCH codes [24]. For the given $n = 1023$ and $k = 923$, all possible redundancy allocation candidates of PBCH codes are presented in Table II.

Fig. 8 shows that $P(\text{decoding failure})$ can be improved by using the side information of \mathbf{s}^+ . If the redundancy for masking l is zero, it means that the side information is ignored. Otherwise, the encoder uses the side information and can improve $P(\text{decoding failure})$. The optimal redundancy allocation (l, r) to minimize $P(\text{decoding failure})$ depends on the ICI S and the additive noise Z . For the given parameters, the optimal redundancy allocation is $(l, r) = (10, 90)$. The pre-read level η_{pre} is set to zero ($\eta_{\text{pre}} = \eta = 0$).

Fig. 9 shows that changing the pre-read level η_{pre} can improve $P(\text{decoding failure})$ significantly. This improvement can be explained by Fig. 10. Compare two threshold voltage distributions of $\eta_{\text{pre}} = 0$ and $\eta_{\text{pre}} = -1$. The threshold voltage distribution of $\eta_{\text{pre}} = -1$ is better than that of $\eta_{\text{pre}} = 0$. The cells whose threshold voltages are between η_{pre} and η are vulnerable to the subsequent ICI and read noise since these cells' threshold voltages are highly probable to be higher than η . By setting $\eta_{\text{pre}} < \eta$, we can also regard these vulnerable cells as stuck-at 1 defects. Thus, all the cells whose threshold voltages are higher than η_{pre} will be written into S_1 , which results in the improvement of the threshold voltage distributions as shown in Fig. 10.



(a) $\sigma_{Z_{\text{read}}} = 0.1$



(b) $\sigma_{Z_{\text{read}}} = 0.3$

Fig. 9. Comparison of $P(\text{decoding failure})$ for different pre-read levels ($\alpha = 0.6, \eta_{\text{pre}} \leq \eta$).

As the gap between η_{pre} and η is bigger, more cells are regarded as stuck-at 1 defects. More defects require more redundancy l for masking defects. If the number of defects is too large, the additive encoding fails to mask defects sufficiently. Thus, $P(\text{decoding failure})$ of $\eta_{\text{pre}} = -2$ is worse than that of $P(\text{decoding failure})$ of $\eta_{\text{pre}} = 0$ as shown in Fig. 9.

By comparing Fig. 9 (a) and Fig. 9 (b), the effect of the read noise can be explained. When $\eta_{\text{pre}} = -1.0$, the optimal redundancy allocation (l, r) in Fig. 9 (a) (for $\sigma_{Z_{\text{read}}} = 0.1$) is $(l, r) = (80, 20)$ and the optimal redundancy allocation in Fig. 9 (b) (for $\sigma_{Z_{\text{read}}} = 0.3$) is $(l, r) = (60, 40)$. It is because more redundancy r for correcting random errors should be allotted as the read noise increases.

V. CONCLUSION

In this paper, the famous examples of Gelfand-Pinsker problem such as “writing on dirty paper” and “memory with defective cells” come together in flash memory. The flash memory channel with the ICI which is similar to the channel of writing on dirty paper has been changed into the model of memory with defective cells by only one pre-read operation. The unique properties of asymmetry between write and erase operations and ICI play a pivotal role in this change. When a

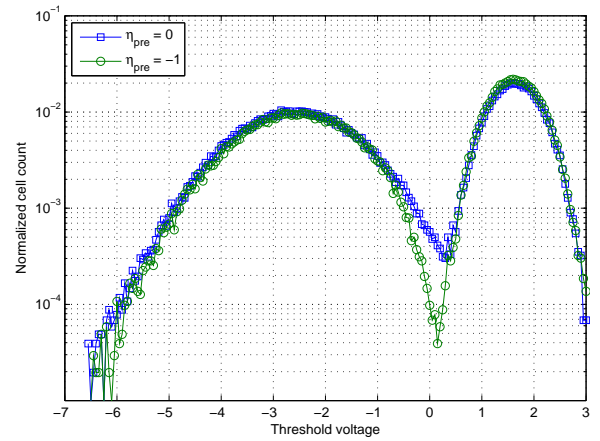


Fig. 10. Comparison of threshold voltage distributions for different pre-read levels η_{pre} ($\alpha = 0.6, \sigma_{Z_{\text{read}}} = 0.3, (l, r) = (100, 0), \eta = 0$).

message is written on dirty flash memory, the dirty spots due to the ICI can be hidden to the reader. Although this paper focused on SLC flash memory, the proposed scheme can be extended to MLC flash memory.

REFERENCES

- [1] K. Prall, “Scaling non-volatile memory below 30nm,” in *Proc. 22nd IEEE Non-Volatile Semiconductor Memory Workshop*, Aug. 2007, pp. 5–10.
- [2] J.-D. Lee, S.-H. Hur, and J.-D. Choi, “Effects of floating-gate interference on NAND flash memory cell operation,” *IEEE Electron Device Lett.*, vol. 23, no. 5, pp. 264–266, May 2002.
- [3] K. Prall and K. Parat, “25nm 64Gb MLC NAND technology and scaling challenges,” in *IEDM Tech. Dig.*, Dec. 2010, pp. 5.2.1–5.2.4.
- [4] K.-T. Park, M. Kang, D. Kim, S.-W. Hwang, B.-Y. Choi, Y.-T. Lee, C. Kim, and K. Kim, “A zeroing cell-to-cell interference page architecture with temporary LSB storing and parallel MSB program scheme for MLC NAND flash memories,” *IEEE J. Solid-State Circuits*, vol. 43, no. 4, pp. 919–928, Apr. 2008.
- [5] N. Shibata, H. Maejima, K. Isobe, K. Iwasa, M. Nakagawa, M. Fujiu, T. Shimizu, M. Honma, S. Hoshi, T. Kawaai, K. Kanebako, S. Yoshikawa, H. Tabata, A. Inoue, T. Takahashi, T. Shano, Y. Komatsu, K. Nagaba, M. Kosakai, N. Motohashi, K. Kanazawa, K. Imamiya, H. Nakai, M. Lasser, M. Murin, A. Meir (Poza), A. Eyal, and M. Shlick, “A 70 nm 16 Gb 16-level-cell NAND flash memory,” *IEEE J. Solid-State Circuits*, vol. 43, no. 4, pp. 929–937, Apr. 2008.
- [6] Y. Li, S. Lee, Y. Fong, F. Pan, T.-C. Kuo, J. Park, T. Samaddar, H. T. Nguyen, M. L. Mui, K. Htoo, T. Kamei, M. Higashitani, E. Yero, G. Kwon, P. Kliza, J. Wan, T. Kaneko, H. Maejima, H. Shiga, M. Hamada, N. Fujita, K. Kanebako, E. Tam, A. Koh, I. Lu, C.-H. Kuo, T. Pham, J. Huynh, Q. Nguyen, H. Chibvongodze, M. Watanabe, K. Oowada, G. Shah, B. Woo, R. Gao, J. Chan, J. Lan, P. Hong, L. Peng, D. Das, D. Ghosh, V. Kalluru, S. Kulkarni, R.-A. Cernea, S. Huynh, D. Pantelakis, C.-M. Wang, and K. Quader, “A 16 Gb 3-bit per cell (x3) NAND flash memory on 56 nm technology with 8 MB/s write rate,” *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 195–207, Jan. 2009.
- [7] G. Dong, N. Xie, and T. Zhang, “On the use of soft-decision error-correction codes in NAND flash memory,” *IEEE Trans. Circuits Syst. I*, vol. 58, no. 2, pp. 429–439, Feb. 2011.
- [8] J. Wang, T. Courtade, H. Shankar, and R. D. Wesel, “Soft information for LDPC decoding in flash: Mutual-information optimized quantization,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Houston, TX, USA, Dec. 2011, pp. 1–6.
- [9] G. Dong, S. Li, and T. Zhang, “Using data postcompensation and predistortion to tolerate cell-to-cell interference in MLC NAND flash memory,” *IEEE Trans. Circuits Syst. I*, vol. 57, no. 10, pp. 2718–2728, Oct. 2010.

- [10] A. Berman and Y. Birk, "Constrained flash memory programming," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, St. Petersburg, Russia, Aug. 2011, pp. 2128–2132.
- [11] Y. Kim, B. V. K. Vijaya Kumar, K. L. Cho, H. Son, J. Kim, J. J. Kong, and J. Lee, "Modulation coding for flash memories," in *Proc. IEEE Int. Conf. Comput., Netw. Commun. (ICNC)*, San Diego, CA, Jan. 2013, pp. 961–967.
- [12] A. Jagmohan, L. A. Lastras-Montano, M. M. Franceschini, M. Sharma, and R. Cheek, "Coding for multilevel heterogeneous memories," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Cape Town, South Africa, May 2010, pp. 1–6.
- [13] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002.
- [14] S. I. Gelfand and M. S. Pinsker, "Coding for channel with random parameters," *Probl. Contr. and Inf. Theory*, vol. 9, no. 1, pp. 19–31, 1980.
- [15] M. H. M. Costa, "Writing on dirty paper," *IEEE Trans. Inf. Theory*, vol. 29, no. 3, pp. 439–441, May 1983.
- [16] A. V. Kuznetsov and B. S. Tsybakov, "Coding in a memory with defective cells," *Probl. Peredachi Inf.*, vol. 10, no. 2, pp. 52–60, Apr.–Jun. 1974.
- [17] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge, U.K.: Cambridge University Press, 2011.
- [18] C. Heegard and A. El Gamal, "On the capacity of computer memory with defects," *IEEE Trans. Inf. Theory*, vol. 29, no. 5, pp. 731–739, Sep. 1983.
- [19] B. S. Tsybakov, "Defect and error correction," *Probl. Peredachi Inf.*, vol. 11, no. 3, pp. 21–30, Jul.–Sep. 1975.
- [20] B. M. Kurkoski, "Rewriting flash memories and dirty-paper coding," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Budapest, Hungary, Jun. 2013, pp. 4353–4357.
- [21] K.-D. Suh, B.-H. Suh, Y.-H. Lim, J.-K. Kim, Y.-J. Choi, Y.-N. Koh, S.-S. Lee, S.-C. Kwon, B.-S. Choi, J.-S. Yum, J.-H. Choi, J.-R. Kim, and H.-K. Lim, "A 3.3 V 32 Mb NAND flash memory with incremental step pulse programming scheme," *IEEE J. Solid-State Circuits*, vol. 30, no. 11, pp. 1149–1156, Nov. 1995.
- [22] Y. Kim, J. Kim, J. J. Kong, B. V. K. Vijaya Kumar, and X. Li, "Verify level control criteria for multi-level cell flash memories and their applications," *EURASIP Journal on Advances in Signal Processing*, vol. 2012, no. 1, pp. 1–13, 2012.
- [23] J. Moon, J. No, S. Lee, S. Kim, S. Choi, and Y. Song, "Statistical characterization of noise and interference in NAND flash memory," *IEEE Trans. Circuits Syst. I*, vol. 60, no. 8, pp. 2153–2164, Aug. 2013.
- [24] C. Heegard, "Partitioned linear block codes for computer memory with "stuck-at" defects," *IEEE Trans. Inf. Theory*, vol. 29, no. 6, pp. 831–842, Nov. 1983.
- [25] B. S. Tsybakov, "Additive group codes for defect correction," *Probl. Peredachi Inf.*, vol. 11, no. 1, pp. 111–113, Jan.–Mar. 1975.
- [26] Y. Kim and B. V. K. Vijaya Kumar, "Coding for memory with stuck-at defects," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Budapest, Hungary, Jun. 2013, pp. 4347–4352.
- [27] —, "Redundancy allocation of partitioned linear block codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, Jul. 2013, pp. 2374–2378.

